

TARDEC

---TECHNICAL REPORT---

No. 14350

By: Wesley Bylsma



CREATION OF VIRTUAL REALITY MODELING LANGUAGE (VRML) APPEARANCE DATA FROM GEOCLR DATA

Distribution: Approved for public release; distribution is unlimited.

U.S. Army Research, Development and Engineering Command
U.S. Army Tank-automotive and Armaments Research
Development and Engineering Center
Detroit Arsenal
6501 East 11 Mile Road
Warren, Michigan 48397-5000

Report Documentation Page			Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 01 NOV 2004		2. REPORT TYPE		3. DATES COVERED -	
4. TITLE AND SUBTITLE CREATION OF VIRTURAL REALITY MODELING LANGUAGE (VRML) APPEARANCE DATA FROM GEOCLR DATA				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) WESLEY BYLSMA				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US ARMY TARDEC, NATIONAL AUTOMOTIVE CENTER,ATTN: AMSRD-TAR-N/MS157,6501 EAST 11 MILE RD,WARREN,MI,48397-5000				8. PERFORMING ORGANIZATION REPORT NUMBER 14350	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT A process for converting Geoclr color files to generic ".appearance" files using the AWK programming language is presented, with the intent of future use in scene assembly into a Virtual Reality Modeling Language (VRML)file.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 9	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

1.0 INTRODUCTION	1
2.0 GEOCLR FORMAT	2
3.0 APPEARANCE FORMAT.....	2
4.0 VRML FORMAT	2
5.0 GAWK CONVERSION.....	3
5.1 INPUT	3
5.2 PARAMETERS	4
5.3 OUTPUT	4
5.4 CODE SECTIONS	4
6.0 SUMMARY/CONCLUSION.....	4
CONTACT	4
REFERENCES	5
DEFINITIONS, ACRONYMS, ABBREVIATIONS.....	5
APPENDIX A – GEO2APPEARANCE.AWK SCRIPT.....	6
APPENDIX B – RECOGNIZED COLOR KEYWORD NAMES FROM SVG 1.1.....	7
FIGURE 1 – ESSENTIAL VRML VISUALIZATION COMPONENTS	1
FIGURE 2 – GAWK PROCESS	3
TABLE 1 - PARAMETER DEFINITIONS	4

CREATION OF VIRTUAL REALITY MODELING LANGUAGE (VRML) APPEARANCE DATA FROM GEOCLR DATA

Wesley Bylsma

U.S. Army Research, Development and Engineering Command (RDECOM)
U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC)
National Automotive Center (NAC)
ATTN: AMSRD-TAR-N/MS157
6501 E 11 Mile Road
Warren, Michigan 48397-5000

1.0 INTRODUCTION

Visualization of complex information is one of the best ways to communicate its meaning. The focus of this effort is on the creation of the appearance portion a Virtual Reality Modeling Language (VRML) file that is used to visualize ground vehicle simulations. As Figure 1 depicts, there are five essential elements that should be included within the composite VRML file for meaningful visualization effects. Only the appearance element is discussed here.

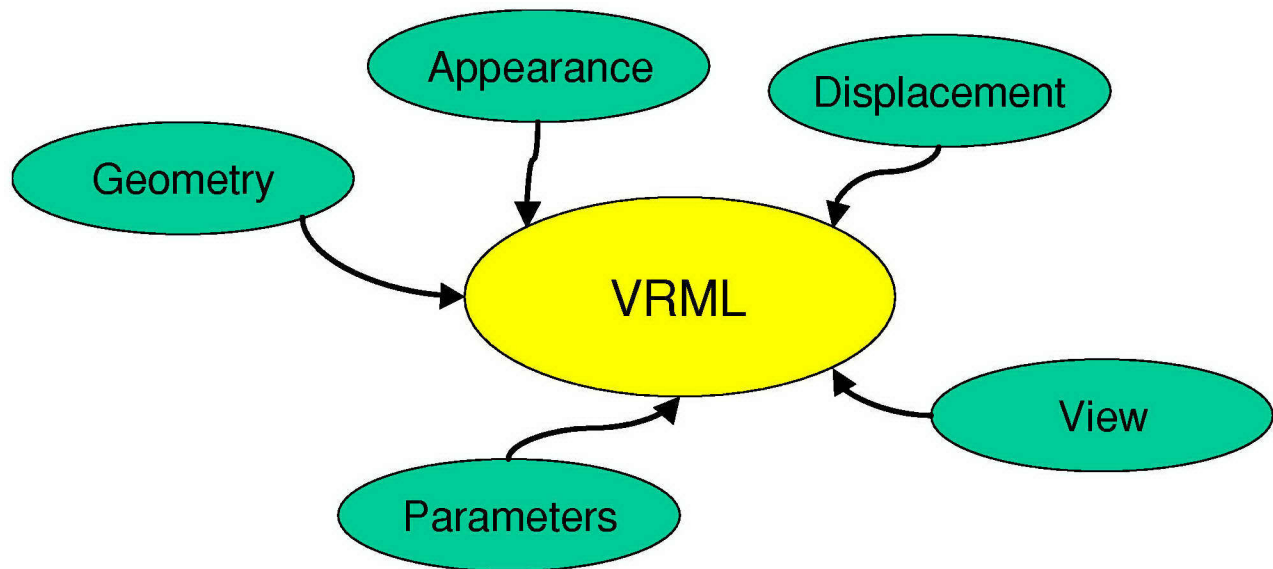


Figure 1 – Essential VRML Visualization Components

The appearance element is a generic file with the ".appearance" extension that contains color, transparency, and texture information for each geometry part. This report addresses the conversion process between the Geoclr format to the generic ".appearance" format to be used with VRML. The conversion process is accomplished with the AWK Programming Language, named after its authors (Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger at Bell Labs), which is designed to provide easy data manipulation and extraction of text files. In this case the Free Software Foundation's GNU version, GAWK, is used. More conversion processes may be developed in the future for inclusion of other appearance file formats. The focus of this discussion is restricted to the conversion of the Geoclr format. Section 2.0 begins with a discussion of the Geoclr format. Section 3.0 discusses the appearance (".appearance") format, section 4.0 discusses the VRML format that will be generated from the ".appearance" format, and section 5.0 outlines the GAWK conversion processes with subsections on specific topics.

2.0 GEOCLR FORMAT

The Geoclr or ".geoclr" file format was developed as an intermediate file format to better associate colors with each geometry Movie.BYU or ".geo" file [1]. The origin of this format is based off of the DADS Interactive Visualizer (DIVA) [5] from the Army High Performance Computing Resource Center (AHPARC). DIVA's usage of geometry and colors is based on ordered lists in separate files (".list" and ".clr"). Each geometry file (Movie.BYU) is associated with RGB colors (values 0 to 255) for each part. For better organization, a separate file with the same name as the geometry file (".geo") was created, but with a different extension (".geoclr"), which defines the color for each part for that geometry. Changing colors for individual parts within a geometry is quicker and less confusing.

The format contains two sections. Below is an example file layout.

```
./geo/vicplsROT.geo
10 100 20
186 189 218
30 30 30
10 100 20
```

The first line specifies the geometry file name. Each following line specifies the RGB color for each part in the geometry listed in the first line. Note the RGB values are in the zero to 255 range as required by DIVA.

3.0 APPEARANCE FORMAT

The appearance file is also a simple ASCII file like the Geoclr file, only with a different format. Its format is laid out into sections encapsulated by a header and trailer line. Below is an example ".appearance" file:

```
# ./geo/vicplsROTb.geoclr
0.1 1 0.2 0 null
1 1 1 0 null
0.3 0.3 0.3 0 null
0.1 1 0.2 0 null
#END
```

The header line begins with a comment "#" and ends with "#END". Following the comment character the name of the Geoclr file it was generated from should be given. Each line between the header and trailer contains appearance information for each part. For each part in the associated Movie.BYU geometry file the RGB color, transparency, and texture file name is specified. Because the original format does not contain transparency or texture information the defaults of "0" and "null" are automatically inserted. Note the value range for RGB colors in VRML is between zero and one, not zero and 255. Also, the transparency value ranges between zero and one with zero being opaque. The texture value is currently a place holder for further development in this area to be included at a later date.

4.0 VRML FORMAT

The appearance data used in the VRML file [2] is included using the Appearance, Material, ImageTexture, and TextureTransform nodes as defined in ISO 14772-1:1997. The ImageTexture and TextureTransform nodes are currently not implemented and will not be discussed in detail here. The creation of these nodes is done during final scene assembly with another program. A description is included here to help understand where the appearance data is included into the final VRML scene file.

The Appearance Node template is

DEF Ax_x Appearance { material ... texture ... textureTransform ... }.

This node defines the appearance of the geometry. By including a name to the node definition, it can be referenced later in other nodes and save space by not requiring a redefinition of all the coordinates. However, separate color definitions are maintained for each part so that transparency and texture features will be unique to them.

The Material Node template is

Material { ambientIntensity ... diffuseColor ... emissiveColor ... shininess ... specularColor ... transparency }

Currently only the following properties are used

Material { diffuseColor ... transparency ... }

This node defines all the color and transparency values. Each part will have an Appearance and Material node. An example section of a VRML (.wrl) file is included below:

```
DEF A9_1 Appearance { #./testd/vicplsROtb.appearance
  material Material {
    diffuseColor 0.100000 1.000000 0.200000
    transparency 0.0
  }
} #./testd/vicplsROtb.appearance
DEF A9_2 Appearance { #./testd/vicplsROtb.appearance
  material Material {
    diffuseColor 1.000000 1.000000 1.000000
    transparency 0.0
  }
} #./testd/vicplsROtb.appearance
DEF A9_3 Appearance { #./testd/vicplsROtb.appearance
  material Material {
    diffuseColor 0.300000 0.300000 0.300000
    transparency 0.0
  }
} #./testd/vicplsROtb.appearance
DEF A9_4 Appearance { #./testd/vicplsROtb.appearance
  material Material {
    diffuseColor 0.100000 1.000000 0.200000
    transparency 0.0
  }
} #./testd/vicplsROtb.appearance
```

5.0 GAWK CONVERSION

The creation of the appearance portion of the VRML file is done with GAWK. This is a very useful scripting language that is available for UNIX and Windows operating systems from The Free Software Foundation ("www.gnu.org/software/gawk/gawk.html") or directly from Bell Labs ("cm.bell-labs.com/cm/cs/awkbook/"). Figure 2 outlines the AWK process.

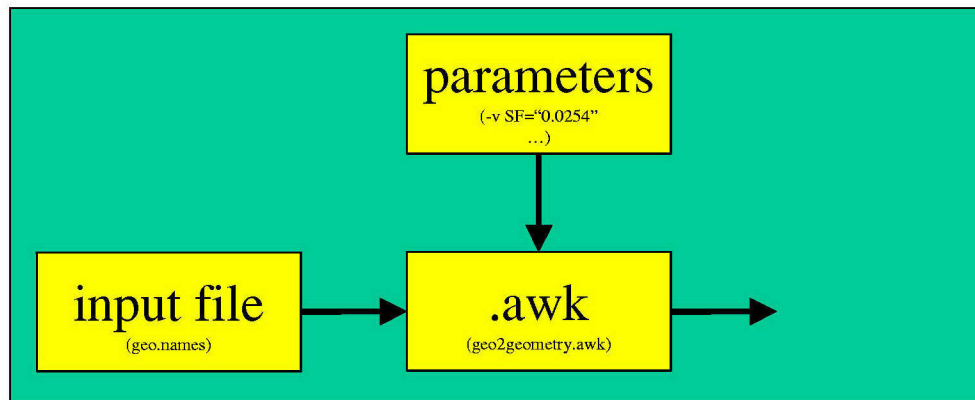


Figure 2 – GAWK process

Parameters are passed to the AWK script before it begins processing the input file. Output can be sent to the standard output or a specified file. An example of the calling structure with the script name "geo2appearance.awk" is given below

```
gawk -v SF=100.0 -v SRC="./geo/" -v SUF=".geoclr" -v DES="./testd/" -f ../src/geo2appearance.awk geoclr.names
```

5.1 INPUT

The input file is named "geoclr.names". Its contents are just a list of files to convert. Its format contains one column. The first column is the Geoclr (or ".geoclr") file name. An example is included below:

```
vicplsROtb
axle1ROtb
axle2ROtb 1
axle34ROtb
axle5ROtb
wheel1ROtb 1
retdrawbar
trailerbitspreadwtank2
RMS1.50-ROT
```

5.2 PARAMETERS

Table 1 defines and describes the parameters passed to the GAWK script for processing the Geoclr files. The VRML file assumes all RGB and transparency values are between zero and one.

Table 1 - Parameter Definitions

VARIABLE	DESCRIPTION	EXAMPLE
SF	Scale Factor. Conversion of RGB (0-255) to RGB (0-1). (If > 1, set to 1.)	100.0
SRC	Source Directory.	"/geo/"
SUF	Geoclr File Suffix Extension.	".geoclr"
DES	Destination Directory.	"/testd/"

It is assumed that all file names listed in the "geoclr.names" input file have the same extension (in the example ".geoclr"). The SF scale factor can be used to "brighten" colors viewed on different platforms since they may be displayed differently.

5.3 OUTPUT

The converted file will have the same name as in the "geoclr.names" file, but with a ".appearance" extension. This format is discussed in section 3.0.

While it is not considered directly part of the conversion process, output of a formatted version of the "geoclr.names" file is required for scene assembly as discussed in Section 1.0 and Figure 1. An example of this simple AWK program is given below

```
gawk '{printf "%3d   %s\n", NR,$1;}' geoclr.names > A.names
```

The "A.names" file is used for visual inspection of the appearance index number to ensure proper alignment with other VRML properties and is used in final scene assembly. An example of the "A.names" file format is given below.

```
1  vicplsROtb
2  axle1ROtb
3  axle2ROtb
5  axle34ROtb
6  axle5ROtb
7  wheel1ROtb
8  retdrawbar
9  trailerbitspreadwtank2
10 RMS1.50-ROT
```

5.4 CODE SECTIONS

The GAWK script "geo2appearance.awk" is included in Appendix A. It begins with checking each name to see if it is a duplicate. If it is, processing skips to the next file name. Processing for each file name is continued as follows:

1. Read header of Geoclr file (Movie.BYU/"geo" file name)
2. Read in RGB color for each part (0-255)
3. If RGB/SF > 1.0 then set RGB value to SF.
4. Write out RGB color for each part (scaled by SF) with default transparency ("0") and image texture ("null") values.

Note that setting the RGB/SF value to the SF if it is greater than one will result in the final RGB value of one when it is scaled for output without using extra storage variables.

6.0 SUMMARY/CONCLUSION

A simple script based conversion process between Geoclr and a generic ".appearance" format was described for use in scene assembly of VRML files. It should be noted that the scene assembly portion mentioned in section 1.0 could be done with X3D [3]. Currently, however, many advanced utilities, such as Cortona Movie Maker [4] will only work with VRML and therefore is the focus at this time.

CONTACT

The author is an engineer with the U.S. Army Research, Development and Engineering Command (RDECOM), located at the U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC). Interested parties can contact the author at the U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC), ATTN: AMSRD-TAR-N/MS157, 6501 E 11 Mile Rd., Warren, Michigan 48397-5000, email: "bylsmaw@tacom.army.mil".

REFERENCES

- [1] Movie.BYU format, "lc.cray.com/doc/movie/".
- [2] The Virtual Reality Modeling Language (VRML), ISO/IEC 14772-1:1997 and 14772-2:2002, "www.web3d.org". (The Virtual Reality Modeling Language consists of two parts. Part 1 (ISO/IEC 14772-1) defines the base functionality and text encoding for VRML. Part 2 (ISO/IEC FDIS 14772-2) defines the base functionality and all bindings for the VRML External Authoring Interface).
- [3] X3D, ISO/IEC Draft 19776-1:200x, 19776-2:200x, 19777:200x, "www.web3d.org". (X3D encodings — ISO/IEC FDIS (Final Draft International Standard) 19776-1:200x (XML encoding) (.html) (.zip 220KB) 2004-09-26 Specifies the encoding of X3D files using the Extensible Markup Language (XML). X3D encodings — ISO/IEC FDIS (Final Draft International Standard) 19776-2:200x (Classic VRML encoding) (.html) (.zip 90KB) 2004-07-21 Specifies the encoding of the functionality and constructs defined in X3D using Classic VRML encoding. X3D language bindings — ISO/IEC FCD (Final Committee Draft) 19777:200x (.html) (.zip 143KB) 2003-05-15 Specifies the binding of the services in the X3D architecture to the ECMAScript programming language for use in X3D internal representation (Script nodes) and for external application access Specifies the binding of the services in the X3D architecture to the Java programming language for use in X3D internal representation (Script nodes) and for external application access).
- [4] Parallelgraphics, Inc. "www.parallelgraphics.com".
- [5] DADS Interactive Visualizer (DIVA), Army High Performance Computing Research Center (AHPCRC), "<http://www.ahpcrc.org/software/diva/>

DEFINITIONS, ACRONYMS, ABBREVIATIONS



RDECOM – U.S. Army Research, Development and Engineering Center
TACOM - U.S. Army Tank-automotive and Armaments Command
TARDEC - TACOM Research, Development and Engineering Center
NAC - National Automotive Center
DADS - Dynamic Analysis and Design System
DIVA - DADS Interactive Visualizer























































































APPENDIX A – GEO2APPEARANCE.AWK SCRIPT





```
#
#geo2appearance
# -v SF=100
# -v SRC="./geo/"
# -v SUF=".geoclr"
# -v DES="./testd/"
#
{
    fi = SRC$1SUF;
    if (fi in names)
    {
        print "Duplicate",fi;
    }
    else
    {
        names[fi]=1;
        fo = DES$1".appearance";
        print "Processing",fi,"->",fo;
        printf "# %s\n",fi > fo;
        err=getline < fi;
        if (err <=0) {print "---error reading",fi; exit;};
        while (getline < fi >0)
        {
            if ($1/SF>1.0) $1=SF;
            if ($2/SF>1.0) $2=SF;
            if ($3/SF>1.0) $3=SF;
            print fo,$1/SF,$2/SF,$3/SF,0,"null"
            print $1/SF,$2/SF,$3/SF,0,"null" >> fo;
        }
        printf "#END\n" >> fo;
        close(fi);
    }
}
```

APPENDIX B – RECOGNIZED COLOR KEYWORD NAMES FROM SVG 1.1

The following is the list of recognized color keywords that can be used as a keyword value for data type <color> from the SVG 1.1 standard:

	aliceblue	rgb(240, 248, 255)		lightpink	rgb(255, 182, 193)
	antiquewhite	rgb(250, 235, 215)		lightsalmon	rgb(255, 160, 122)
	aqua	rgb(0, 255, 255)		lightseagreen	rgb(32, 178, 170)
	aquamarine	rgb(127, 255, 212)		lightskyblue	rgb(135, 206, 250)
	azure	rgb(240, 255, 255)		lightslategray	rgb(119, 136, 153)
	beige	rgb(245, 245, 220)		lightslategray	rgb(119, 136, 153)
	bisque	rgb(255, 228, 196)		lightsteelblue	rgb(176, 196, 222)
	black	rgb(0, 0, 0)		lightyellow	rgb(255, 255, 224)
	blanchedalmond	rgb(255, 235, 205)		lime	rgb(0, 255, 0)
	blue	rgb(0, 0, 255)		limegreen	rgb(50, 205, 50)
	blueviolet	rgb(138, 43, 226)		linen	rgb(250, 240, 230)
	brown	rgb(165, 42, 42)		magenta	rgb(255, 0, 255)
	burlywood	rgb(222, 184, 135)		maroon	rgb(128, 0, 0)
	cadetblue	rgb(95, 158, 160)		mediumaquamarine	rgb(102, 205, 170)
	chartreuse	rgb(127, 255, 0)		mediumblue	rgb(0, 0, 205)
	chocolate	rgb(210, 105, 30)		mediumorchid	rgb(186, 85, 211)
	coral	rgb(255, 127, 80)			

	cornflowerblue	rgb(100, 149, 237)		mediumpurple	rgb(147, 112, 219)
	cornsilk	rgb(255, 248, 220)		mediumseagreen	rgb(60, 179, 113)
	crimson	rgb(220, 20, 60)		mediumslateblue	rgb(123, 104, 238)
	cyan	rgb(0, 255, 255)		mediumspringgreen	rgb(0, 250, 154)
	darkblue	rgb(0, 0, 139)		mediumturquoise	rgb(72, 209, 204)
	darkcyan	rgb(0, 139, 139)		mediumvioletred	rgb(199, 21, 133)
	darkgoldenrod	rgb(184, 134, 11)		midnightblue	rgb(25, 25, 112)
	darkgray	rgb(169, 169, 169)		mintcream	rgb(245, 255, 250)
	darkgreen	rgb(0, 100, 0)		mistyrose	rgb(255, 228, 225)
	darkgrey	rgb(169, 169, 169)		moccasin	rgb(255, 228, 181)
	darkkhaki	rgb(189, 183, 107)		navajowhite	rgb(255, 222, 173)
	darkmagenta	rgb(139, 0, 139)		navy	rgb(0, 0, 128)
	darkolivegreen	rgb(85, 107, 47)		oldlace	rgb(253, 245, 230)
	darkorange	rgb(255, 140, 0)		olive	rgb(128, 128, 0)
	darkorchid	rgb(153, 50, 204)		olivedrab	rgb(107, 142, 35)
	darkred	rgb(139, 0, 0)		orange	rgb(255, 165, 0)
	darksalmon	rgb(233, 150, 122)		orangered	rgb(255, 69, 0)
	darkseagreen	rgb(143, 188, 143)		orchid	rgb(218, 112, 214)
	darkslateblue	rgb(72, 61, 139)		palegoldenrod	rgb(238, 232, 170)
	darkslategray	rgb(47, 79, 79)		palegreen	rgb(152, 251, 152)
	darkslategrey	rgb(47, 79, 79)		paleturquoise	rgb(175, 238, 238)
	darkturquoise	rgb(0, 206, 209)		palevioletred	rgb(219, 112, 147)
	darkviolet	rgb(148, 0, 211)		papayawhip	rgb(255, 239, 213)
	deeppink	rgb(255, 20, 147)		peachpuff	rgb(255, 218, 185)
	deepskyblue	rgb(0, 191, 255)		peru	rgb(205, 133, 63)
	dimgray	rgb(105, 105, 105)		pink	rgb(255, 192, 203)
	dimgrey	rgb(105, 105, 105)		plum	rgb(221, 160, 221)
	dodgerblue	rgb(30, 144, 255)		powderblue	rgb(176, 224, 230)
	firebrick	rgb(178, 34, 34)		purple	rgb(128, 0, 128)
	floralwhite	rgb(255, 250, 240)		red	rgb(255, 0, 0)
	forestgreen	rgb(34, 139, 34)		rosybrown	rgb(188, 143, 143)
	fuchsia	rgb(255, 0, 255)		royalblue	rgb(65, 105, 225)
	gainsboro	rgb(220, 220, 220)		saddlebrown	rgb(139, 69, 19)
	ghostwhite	rgb(248, 248, 255)		salmon	rgb(250, 128, 114)
	gold	rgb(255, 215, 0)		sandybrown	rgb(244, 164, 96)
	goldenrod	rgb(218, 165, 32)		seagreen	rgb(46, 139, 87)
	gray	rgb(128, 128, 128)		seashell	rgb(255, 245, 238)
	grey	rgb(128, 128, 128)		sienna	rgb(160, 82, 45)
	green	rgb(0, 128, 0)		silver	rgb(192, 192, 192)
	greenyellow	rgb(173, 255, 47)		skyblue	rgb(135, 206, 235)
	honeydew	rgb(240, 255, 240)		slateblue	rgb(106, 90, 205)
	hotpink	rgb(255, 105, 180)		slategray	rgb(112, 128, 144)
	indianred	rgb(205, 92, 92)		slategrey	rgb(112, 128, 144)
	indigo" data-bbox="71 667 85 678"/>	rgb(75, 0, 130)		snow	rgb(255, 250, 250)
	ivory	rgb(255, 255, 240)		springgreen	rgb(0, 255, 127)
	khaki	rgb(240, 230, 140)		steelblue	rgb(70, 130, 180)
	lavender	rgb(230, 230, 250)		tan	rgb(210, 180, 140)
	lavenderblush	rgb(255, 240, 245)		teal	rgb(0, 128, 128)

	lawngreen	rgb(124, 252, 0)		thistle	rgb(216, 191, 216)
	lemonchiffon	rgb(255, 250, 205)		tomato	rgb(255, 99, 71)
	lightblue	rgb(173, 216, 230)		turquoise	rgb(64, 224, 208)
	lightcoral	rgb(240, 128, 128)		violet	rgb(238, 130, 238)
	lightcyan	rgb(224, 255, 255)		wheat	rgb(245, 222, 179)
	lightgoldenrodyellow	rgb(250, 250, 210)		white	rgb(255, 255, 255)
	lightgray	rgb(211, 211, 211)		whitesmoke	rgb(245, 245, 245)
	lightgreen	rgb(144, 238, 144)		yellow	rgb(255, 255, 0)
	lightgrey	rgb(211, 211, 211)		yellowgreen	rgb(154, 205, 50)